Requirement Engineering of Software Product Lines: Extracting Variability using NLP

Alessandro Fantechi
Dipartimento di Ingegneria dell'Informazione
Università di Firenze, Italy
Email: alessandro.fantechi@unifi.it

Alessio Ferrari Stefania Gnesi
Istituto di Scienza e Tecnologie
dell'Informazione "A.Faedo",
ISTI-CNR, Pisa, Italy
Email: {alessio.ferrari, stefania.gnesi}@isti.cnr.it

Laura Semini
Dipartimento di Informatica
Università di Pisa, Italy
Email: semini@di.unipi.it

***Abstract*—The engineering of software product lines begins with the identification of the possible variation points. To this aim, natural language (NL) requirement documents can be used as a source from which variability-relevant information can be elicited. In this paper, we propose to identify variability issues as a subset of the ambiguity defects found in NL requirement documents. To validate the proposal, we single out ambiguities using an available NL analysis tool, QuARS, and we classify the ambiguities returned by the tool by distinguishing among false positives, real ambiguities, and variation points, by independent analysis and successive agreement phase. We consider three different sets of requirements and collect the data that come from the analysis performed.**

I. Introduction

Among the fundamental activities of software product line engineering (SPLE) there is the identification of the variability in different artifacts of the system, such as requirements, architecture and test cases [1]. In particular, in the requirement engineering of software product lines (SPL), several researches have focused on exploiting natural language processing (NLP) techniques and tools to extract information related to features and variability from documents in natural language (NL) in general [2], [3], [4] and requirement documents in particular [5], [6].

In previous works [7], [8], we initiated a line of research to extract variability issues from the NL sentences in a requirement document using an existing NLP tool aimed at revealing *ambiguity* defects. The underlying intuition is that often ambiguity in requirements is due to the (conscious or subconscious) need to postpone choices for later decisions in the implementation of the system. Ambiguities normally cause inconsistencies between the expectation of the customer and the product developed, and possibly lead to undesirable reworks on the artifacts. However, ambiguity can also be used as a way to capture variability aspects to be solved later in the software development.

Specifically, we envisioned an approach to achieve automated support to variability elicitation by analysing the outcomes of automated ambiguity detection applied to some set of requirements by means of the QuARS (Quality Analyser for Requirements Specifications) tool [9], [10], a well known tool for analyzing NL requirements in a systematic and automatic way. QuARS allows to perform an initial parsing of NL requirements for detecting potential linguistic defects that can determine interpretation problems.

In [7] we started bottom up, and we focused on the correspondence between ambiguity indicators and feature model fragments, to define a systematic way of building a feature model from a set of variability points extracted with a NLP tool. Then, in [8] we initiated a validation of the approach with QuARS, in order to:

- assess whether ambiguities in NL requirements can be considered as potential variation points,
- assess to which extent the process of variability identification can be automated with an ambiguity detection tool such as QuARS,
- refine the definition of a possible NLP tool for variability detection.

To this aim, our research questions (RQs) are the following:

RQ1 Is automated ambiguity detection in NL requirement documents a proxy to detect variability?

RQ2 Which are the ambiguity indicators that are most relevant to detect variability, if any?

RQ3 The variability detection process can benefit from adding a new variability tailored dictionay to an existing NL requirements analysis tool?

Some initial answers to theses RQs are in [8], where we have shown that an ambiguity detection tool can be helpful to detect variability, stimulating a more systematic and deep experience.

Hence, we here systematically address the RQs using three requirement documents of medium to large size, that are publicly available and refer to different domains and systems to be built, and assess the results by using well known statistical measures. Exploiting the experience made in past work, we also define a new dictionary to be added to QuARS, designed to provide an indicator for variability, and we extend the experiments and measures to this new indicator as well.

The remainder of the paper is structured as follows. In Sect. II we make explicit the relevance of our work to RE. In Sect. III we present the main features of QuARS. In Sect. IV we present our research design. In Sect. V and VI we discuss the results and threats to validity. Sect. VII concludes the paper.

II. Relevance to RE

In the field of product line engineering, the use of NLP techniques to identify variation points in single requirements documents is justified by the following observations:

(1) NL is intrinsically ambiguous, and this is seen often as a possible source of problems in the later interpretation of requirements. However, ambiguity or underspecification at requirements level can in some cases give an indication of possible variability, either in design choices, in implementation choices or configuration. The intuition is that often ambiguity in requirements may be due to the need to postpone choices for later decisions in the implementation of the system [11], [12], [7].

(2) Taking into account the results of our previous analyses conducted on different requirements documents with NLP analysis tools, a first classification of the forms of ambiguity that indicate variation points has been proposed starting from the analysis of documents describing real systems [7], [8]. In this paper we presents an extensive, and empirically grounded, experimentation to validate the approach presented in our previous works [7], [8], which had an exploratory nature and were mainly oriented to scope the problem.

III. QuARS

QuARS was introduced as an automatic analyzer of requirement documents [10]. QuARS performs an initial parsing of NL requirements for automatic detection of potential linguistic defects that can determine ambiguity problems impacting the following development stages. QuARS executes a linguistic analysis of a requirements document in plain text format and points out the sentences that are defective according to the quality model described in [13]. The defect identification process is split in two parts: (i) the "lexical analysis" capturing *optionality, subjectivity, vagueness, multiplicity and weakness* defects, by identifying candidate defective terms that are identified into a corresponding set of dictionaries; and (ii) the "syntactical analysis" capturing *implicity and underspecification* defects. In the same way, detected defects may however be *false defects* – i.e., false positives. In Table I we present the indicators used by QuARS to detect lexical and syntactical ambiguity defects in NL sentences.

In this work, we also exploit the capability of QuARS to add dictionaries for new indicators: in Table II we present the new dictionary collecting potential variability related terms (Tailored Dictionary). These terms were identified during the analysis of the requirements to answer RQ1 and RQ2 and they will be used to answer to RQ3.

IV. Research Design

A. Requirement Documents

We base our analysis on three requirements documents that are very different from each other in terms of domain, characteristics of the systems, and background and experience of of their authors.

They can be downloaded from the PURE requirements dataset [14], available at http://fmt.isti.cnr.it/nlreqdataset/ (file names: *2007-EireneFun*, *2009-library* and *2010-blit*), and are described below. The documents in textual format, together with the annotations performed on them, are available at: https://github.com/isti-fmt-nlp/AmbiguityToVariability.

1) European Integrated Railway Radio Enhanced Network (Eirene): The "Eirene" document has been issued by the GSM-R Functional Group and it specifies the functional requirements for a digital radio standard for the European railways. The document includes 475 requirements.

2) Library: The second document, "Library", was prepared by the Galecia Group, a company specialised in libraries and organizations supporting libraries. It describes the functional and nonfunctional requirements for the System Administration Module of the Integrated Library System of a urban library system. It includes 94 requirements.

3) Blit: "Blit", is a draft of the functional specification of the requirements of a business project management tool, required by a company for re-writing its core Laboratory Information System to improve the performance. The authors of the document, which includes 55 requirements, are anonymous.

B. Data Collection and Analysis

To address RQ1 and RQ2 we first analyse, using QuARS, the three requirement documents described in Sect. IV-A according to all the indicators given in Table I. Then, to elicit the potential variability hidden in a requirement document, we perform an assessment of the output of the tool, for each ambiguity indicator, aimed at classifying the defective sentences and distinguish among: false positives, variability points, and actual ambiguities.

Finally, to answer to RQ3, we repeat the same procedure using the Tailored Dictionary (Table II), to determine the appropriateness of this newly introduced set of terms.

More specifically, the *data collection* procedure, for each document, consists of the following steps:

Automatic Detection: The document is given as input to QuARS in textual format, and QuARS produces a set of sentences that are considered ambiguous, together with the term or expression that is the source of the ambiguity;

Review: The output of QuARS is reviewed by the 3rd author on the one side, and, independently, by the 4th one. Each defect identified by QuARS has been identified as: false positives, variability, or ambiguity;

Assessment: The classification is reviewed by the 3rd and 4th authors, who discussed the discrepancies emerged in the judgment and looked for an agreement. The classification derived in this phase is the one used for data analysis.

Review and assessment phases, that highlight variation points, are based on the criteria introduced in our previous paper [8]: ambiguity in requirements may be due to the need to enlighten possible variation points in an early phase of software and system development and to postpone choices for

Sub-characteristic	*Indicators*
Vagueness	The occurrence of Vagueness-revealing terms (e.g.: clear, easy, strong, good, bad, useful, significant, adequate, recent,…) is considered a vagueness indicator
Subjectivity	The occurrence of Subjectivity-revealing wordings (e.g.: similar, similarly, having in mind, take into account, as [adjective] as possible,…) is considered a subjectivity indicator
Optionality	The occurrence of Optionality-revealing wordings (e.g.: possibly, eventually, case, if possible, if appropriate, if needed,…) is considered an optionality indicator
Implicity	The occurrence of: • Subjects or complements expressed by means of: Demonstrative adjectives (this, these, that, those) or Pronouns (it, they...) or • Terms having the determiner expressed by a demonstrative adjective (this, these, that, those) or implicit adjective (e.g. :previous, next, following, last...) or preposition (e.g.: above, below...) is considered an implicity indicator
Weakness	The occurrence of Weak verbs (e.g.: can, could, may,…) is considered a weakness indicator
Under-specification	The occurrence of wordings needing to be instantiated (e.g.: information, interface, that must be better defined, flow instead of data flow, control flow, access instead of write access, remote access, authorized access, testing instead of functional testing, structural testing, unit testing, etc.) is considered an under-specification indicator.
Multiplicity	The occurrence of multiplicity-revealing terms: and/or, or, ... is considered a multiplicity indicator, as well as the presence of itemized lists.

TABLE I
QuARS AMBIGUITY INDICATORS

Sub-characteristic	*Indicators*
Variability	The occurrence of variability-revealing terms: if, where, whether, when, choose, choice, implemented, implement, implements, provided, provide, provides, available, feature, range, select, selected, selects, configurable, configurate, configurated, configurates

TABLE II
QuARS NEW TAILORED DICTIONARY

later decisions in the implementation of the system. Hence, the analysis of the defective requirements is guided by the general question “Can this requirement hide a variation point?”. More concrete review guidelines may depend on the QuARS indicators. In the cases of *implicity* and *subjectivity*, there is no intuition that a defect can actually be a variation point, and the analysis is only based on the judgment of the reviewer. With *under-specification* and *vagueness* the criterion to identify a variability is the existence of more than one possible instance of the defective term. With *multiplicity* the reviewer can mark as false positives all the requirements where the term “or”, “and/or” and similar relate two sentences or two adjectives, and use their judgment in the cases where they relate nouns. The cases of *weakness* and *optionality* are treated similarly, since the nature of these defects is inherently associated to variation points, especially when they appear in functional requirements [8].

The *data analysis* procedure, for each document, consists of the following steps:

Agreement: The annotations of the two independent reviewers are compared by means of the Fleiss Kappa [15], to have an indication of judgments’ discrepancies.

Quantitative Analysis: The number of defects found by the tool (fnd), false positives (fp), variability indicators (var), and the actual ambiguities (amb) is computed for each indicator, based on the data collected in the Assessment step.

With this data we can assess the appropriateness of QuARS to answer to our research questions. To this end we use *precision* i.e. the fraction of relevant instances among the retrieved instances. Precision aims at answering RQ1, and to give a broad view about the indicators that are more relevant for variability detection (RQ2).

Precision general formula is:

$$prec_i = \frac{tp_i}{fnd_i}$$

where $fnd_i = tp_i + fp_i$ is the number of all the terms found by QuARS, i.e. the sum of true positive and false positive (resp.) classifications for indicator i. To have a full picture and to compare the results, we calculate the precision of QuARS in detecting, for each indicator, both ambiguities and variabilities:

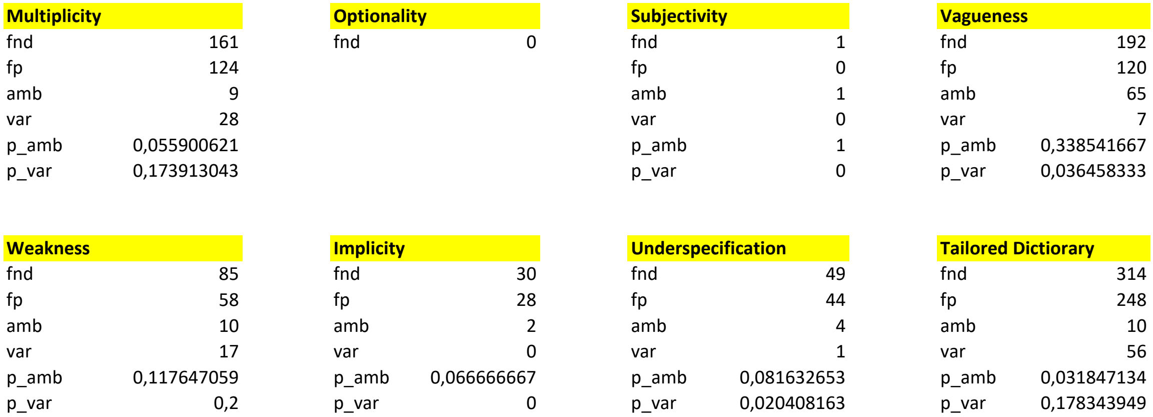
$$p\_amb_i = \frac{amb_i}{fnd_i} \qquad p\_var_i = \frac{var_i}{fnd_i}$$

Aggregate values are also computed for these indicators.

Qualitative Analysis: For each indicator showing a substantially higher precision than others, we look whether some terms in the corresponding vocabulary can be considered variability-related terms. This analysis aims to provide a more refined answer to RQ2. Furthermore, we inspect the requirements to discover additional terms to be used, provide an answer to RQ3, and further iterate the data analysis process.

V. RESULTS AND OBSERVATIONS

The agreement between annotators on the whole output of QuARS resulted in 0.455, indicating a moderate agreement [16]. Figure 1 presents the results of the quantitative analysis for the seven QuARS indicators and for the new Tailored Dictionary. For each indicator, we report the number of defects found by the tool (fnd), the number of false positives



Multiplicity	
fnd	161
fp	124
amb	9
var	28
p_amb	0,055900621
p_var	0,173913043

Optionality	
fnd	0

Subjectivity	
fnd	1
fp	0
amb	1
var	0
p_amb	1
p_var	0

Vagueness	
fnd	192
fp	120
amb	65
var	7
p_amb	0,338541667
p_var	0,036458333

Weakness	
fnd	85
fp	58
amb	10
var	17
p_amb	0,117647059
p_var	0,2

Implicity	
fnd	30
fp	28
amb	2
var	0
p_amb	0,066666667
p_var	0

Underspecification	
fnd	49
fp	44
amb	4
var	1
p_amb	0,081632653
p_var	0,020408163

Tailored Dictionary	
fnd	314
fp	248
amb	10
var	56
p_amb	0,031847134
p_var	0,178343949

Fig. 1. Aggregated results of the quantitative analysis

(fp), the actual ambiguities (amb), and the variability indicators (var), as classified by manual inspection. The last two rows indicate the precision of QuARS in identifying ambiguities (p_amb) and variability points (p_var), respectively.

Results in Figure 1 are aggregated, and do not distinguish among the different requirement documents. We comment on these results since the collected data were uniform with respect to the different documents. For completeness, we provide detailed results in Figure 2.

We analyse in the following the single indicators.

With *multiplicity*, variability is actually an option and true positives are equally distributed between "or" and "and/or". An example is:

> The setting up and closing down of a multi-driver call shall be simplified using automation or guidance through the steps required.

We did not find any *optionality* defect in any of the considered documents, and only 1 for *subjectivity*: at least for these case studies these indicators are definitely not relevant.

Vagueness is due to the presence of undetermined adjectives and adverbs and can mask a variability, even though in few cases, like in:

> Various types of call restriction may be employed by the railways as an additional security measure.

We classified as variability sources some requirements including the terms *possible* and *capable*. However, the true source of variablity was due to term *where*:

> Where fax functionality is provided, it shall be possible to interrupt the fax to make or receive a high priority voice or data call.

We have exploited this observation to add to the Tailored Dictionary terms as *where, when, if, whether*.

A further ambiguity indicator of QuARS is *weakness*. A sentence with verbs *may* or *can*, etc. is considered weak, e.g.:

> A driver may be provided with a handheld portable to allow communications whilst the driver is outside the train.

Besides, sometimes requirements are explicitely labelled with a *may/can* to indicate that their implementation is optional, so introducing a variability. A good percentage of the defective sentences revealed by QuARS expresses optional requirements, as shown in Figure 1. We also report that variablity is due to *may* much more than to *can*.

For the *implicity* indicator, a sentence is considered defective if its subject or complements are *implicit*, being expressed by demonstrative adjectives (this, these, that, those) or pronouns (it, they, *etc.*) instead of by a noun. In the considered documents, implicity is in most cases resolved when reading the sentence, and, in any case, it is never an indication of possible variability. Also in the case of *under-specification*, most defective sentences are false positives, and almost no variability is hidden behind. An example of variablity is:

> It shall be possible to present radio related prompts and information in a number of different languages.

Finally, the defects revealed using the Tailored Dictionary show a performance in line with that related to multiplicity and weakness indicators. However, we observed that, in most cases, variability was hidden in requirements including pairs of terms of the kind *when. . . possible*, *if. . . implemented*, etc, when occurring in the same sentence, as in:

> If the text message facility is implemented, it shall not interfere with the ability of users to make or receive high priority voice or data calls.

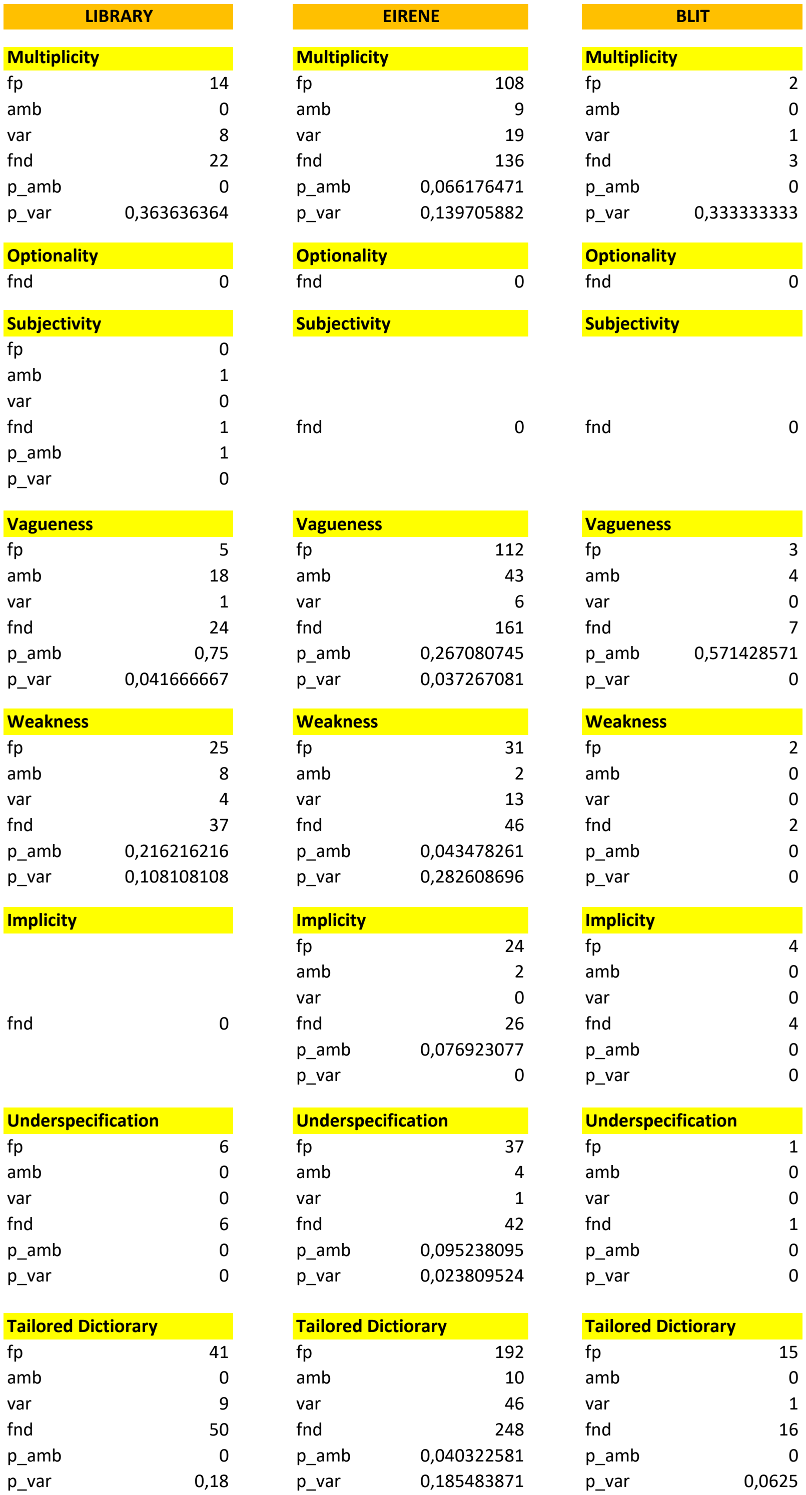
while they were most probably false positives when occurring in two different sentences of the same requirement, as in:

> When the call is connected to the controller, an audible and visual indication is to be provided to the driver.

To identify these cases, we plan to empower QuARS with a pattern-based approach similar to the one that we applied in [17].

VI. Threats to validity

Construct Validity An objective and widely used metric, i.e., precision, was used to assess the appropriateness of QuARS



LIBRARY		EIRENE		BLIT	
Multiplicity		**Multiplicity**		**Multiplicity**	
fp	14	fp	108	fp	2
amb	0	amb	9	amb	0
var	8	var	19	var	1
fnd	22	fnd	136	fnd	3
p_amb	0	p_amb	0,066176471	p_amb	0
p_var	0,363636364	p_var	0,139705882	p_var	0,333333333
Optionality		**Optionality**		**Optionality**	
fnd	0	fnd	0	fnd	0
Subjectivity		**Subjectivity**		**Subjectivity**	
fp	0				
amb	1				
var	0				
fnd	1	fnd	0	fnd	0
p_amb	1				
p_var	0				
Vagueness		**Vagueness**		**Vagueness**	
fp	5	fp	112	fp	3
amb	18	amb	43	amb	4
var	1	var	6	var	0
fnd	24	fnd	161	fnd	7
p_amb	0,75	p_amb	0,267080745	p_amb	0,571428571
p_var	0,041666667	p_var	0,037267081	p_var	0
Weakness		**Weakness**		**Weakness**	
fp	25	fp	31	fp	2
amb	8	amb	2	amb	0
var	4	var	13	var	0
fnd	37	fnd	46	fnd	2
p_amb	0,216216216	p_amb	0,043478261	p_amb	0
p_var	0,108108108	p_var	0,282608696	p_var	0
Implicity		**Implicity**		**Implicity**	
		fp	24	fp	4
		amb	2	amb	0
		var	0	var	0
fnd	0	fnd	26	fnd	4
		p_amb	0,076923077	p_amb	0
		p_var	0	p_var	0
Underspecification		**Underspecification**		**Underspecification**	
fp	6	fp	37	fp	1
amb	0	amb	4	amb	0
var	0	var	1	var	0
fnd	6	fnd	42	fnd	1
p_amb	0	p_amb	0,095238095	p_amb	0
p_var	0	p_var	0,023809524	p_var	0
Tailored Dictiorary		**Tailored Dictiorary**		**Tailored Dictiorary**	
fp	41	fp	192	fp	15
amb	0	amb	10	amb	0
var	9	var	46	var	1
fnd	50	fnd	248	fnd	16
p_amb	0	p_amb	0,040322581	p_amb	0
p_var	0,18	p_var	0,185483871	p_var	0,0625

Fig. 2. Detailed results of the quantitative analysis

in identifying variabilities. Since the data used to calculate the precision are based on subjective evaluations, to mitigate subjectivity threats, we assessed the agreement among annotators by means of the Fleiss Kappa, which indicated a moderate agreement. Similarly to other works [18], we arguably consider this value acceptable in the inherently ambiguous context of NL-RE. Therefore, the reviewing guidelines provided in Sect. IV-B can be considered as sufficiently detailed. Furthermore, an Assessment followed the Review step to define the final dataset used for analysis. It is worth noting that we did not use the recall measure to assess our results, since we annotated the requirements for variability after the application of QuARS. Therefore, some variability indicators may exist in the original requirements that are not considered in our analysis. This threat was not specifically mitigated in this work, also because with our RQs we did not aim to compare manual and automatic assessments, but we were oriented to investigate how to better exploit the combination of NLP and human analysis.

Internal Validity The main threat to the internal validity of the study is the involvement of the authors of this work in the Review and Assessment phase of the data collection procedure. We agree that the researcher bias might have played a role in these phases. However, this is mitigated by the evaluation of agreement. Furthermore, other researchers can replicate our approach using the publicly available[1] QuARS tool, and using the documents employed in our evaluation.

External Validity Our results are limited to three requirements documents. However, the documents are representative of different domains, and we have observed that several variability-related terms are common among the documents. Similar results were obtained in our previous study [8]. Therefore, we argue that our study has the potential to be generalised to other domains, and other requirements documents.

VII. Conclusion

Ambiguities in NL requirements may hide indications of variability. In our previous work [8], we proposed to use QuARS, an ambiguity detection tool, to identify terms that reveal variability. In this paper, we present the empirical evaluation of the idea, performed on three documents from industry of 624 requirements in total. The evaluation shows that some ambiguous terms, such as "and/or" "or", and weak terms such as "may" or "could" are more likely to indicate variability rather than ambiguity. Instead, typically vague terms, such as "useful", "significant", etc. are more likely to indicate ambiguity. We also show that rule-based lexical approaches such as the one mainly used by QuARS lead to a relevant number of false positive cases. This indicates that, although other authors recommend the usage of lexical approaches for RE activities to maximise recall [19], there is also a need for more advanced syntactic/semantic solutions to increase precision, and make NLP for RE approaches applicable in practice. This is particularly true for variability detection: indeed, while ambiguities in safety-critical requirements may lead to potential system failures, non-identified variabilities may not have such severe consequences, and loosing some recall in favour of precision would speed-up the system analysis process. Specifically, the authors employed about 20 hours to identify variabilities and ambiguities, which resulted to be about the 25% of the total number of terms identified by QuARS. A tool with 100% precision would have reduced the work to five hours.

References

[1] L. Chen, M. Ali Babar, and N. Ali, "Variability management in software product lines: a systematic review," in *SPLC'13*. Carnegie Mellon University, 2009, pp. 81–90.

[2] S. B. Nasr, G. Bécan, M. Acher, J. B. Ferreira Filho, N. Sannier, B. Baudry, and J.-M. Davril, "Automated extraction of product comparison matrices from informal product descriptions," *JSS*, vol. 124, pp. 82–103, 2017.

[3] A. Ferrari, G. O. Spagnolo, and F. Dell'Orletta, "Mining commonalities and variabilities from natural language documents," in *SPLC'13*. ACM, 2013, pp. 116–120.

[4] Y. Li, S. Schulze, and G. Saake, "Reverse engineering variability from natural language documents: A systematic literature review," in *SPLC'17*. ACM, 2017, pp. 133–142.

[5] N. Itzik, I. Reinhartz-Berger, and Y. Wand, "Variability analysis of requirements: Considering behavioral differences and reflecting stakeholders perspectives," *IEEE Transactions on Software Engineering*, vol. 42, no. 7, pp. 687–706, 2016.

[6] N. H. Bakar, Z. M. Kasirun, and N. Salleh, "Feature extraction approaches from natural language requirements for reuse in software product lines: A systematic literature review," *JSS*, vol. 106, pp. 132–149, 2015.

[7] A. Fantechi, S. Gnesi, and L. Semini, "Ambiguity defects as variation points in requirements," in *VAMOS '17*. New York, NY, USA: ACM, 2017, pp. 13–19.

[8] A. Fantechi, A. Ferrari, S. Gnesi, and L. Semini, "Hacking an ambiguity detection tool to extract variation points: an experience report," in *VAMOS'18*. ACM, 2018, pp. 43–50.

[9] F. Fabbrini, M. Fusani, S. Gnesi, and G. Lami, "An automatic quality evaluation for natural language requirements," in *REFSQ'01*, vol. 1, 2001, pp. 4–5.

[10] S. Gnesi, G. Lami, and G. Trentanni, "An automatic tool for the analysis of natural language requirements," *Comput. Syst. Sci. Eng.*, vol. 20, no. 1, 2005.

[11] H. Yang, A. De Roeck, V. Gervasi, A. Willis, and B. Nuseibeh, "Speculative requirements: Automatic detection of uncertainty in natural language requirements," in *RE'12*. IEEE, 2012, pp. 11–20.

[12] A. Ferrari, P. Spoletini, and S. Gnesi, "Ambiguity as a resource to disclose tacit knowledge," in *RE'15*. IEEE, 2015, pp. 26–35.

[13] D. M. Berry, A. Bucchiarone, S. Gnesi, G. Lami, and G. Trentanni, "A new quality model for natural language requirements specifications," in *REFSQ'06*, 2006, pp. 115–128.

[14] A. Ferrari, G. O. Spagnolo, and S. Gnesi, "PURE: A dataset of public requirements documents," in *RE'17*. IEEE, 2017, pp. 502–505.

[15] J. L. Fleiss, B. Levin, and M. C. Paik, *The measurement of interrater agreement*. Wiley Online Library, 2004.

[16] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *biometrics*, pp. 159–174, 1977.

[17] B. Rosadini, A. Ferrari, G. Gori, A. Fantechi, S. Gnesi, I. Trotta, and S. Bacherini, "Using NLP to detect requirements defects: An industrial experience in the railway domain," in *REFSQ'17*. Springer, 2017, pp. 344–360.

[18] H. Femmer, D. M. Fernández, S. Wagner, and S. Eder, "Rapid quality assurance with requirements smells," *JSS*, vol. 123, pp. 190–213, 2017.

[19] S. F. Tjong and D. M. Berry, "The design of SREE - a prototype potential ambiguity finder for requirements specifications and lessons learned," in *REFSQ'13*. Springer, 2013, pp. 80–95.

[1] Actually, the tool is provided upon request to the 3rd author: this allows us to keep track of the users of the tool, and to receive feedback on its usage.